

### **REMARKS**

This Amendment is filed concurrently with a Request for Continued Examination (RCE) and the fee for a two-month extension of time, large entity.

Claims 1-5, 8-12, 63-65 and 108 stand rejected as being unpatentable over Mockapetris in view of Nguyen. Claim 109 was further rejected as being unpatentable over the Mockapetris-Nguyen combination, in further view of San Andres. Reconsideration and withdrawal of these rejections are respectfully requested.

The Amendment After Final Rejection of February 3, 2011 was entered (for Appeal purposes only) for the reasons detailed in the Advisory Action of March 4, 2011. Specifically, the Examiner stated that Mockapetris

**“directly sends information to the first server and indirectly sends information to each subsequent server in the ring. But for Mockapetris starting the process, none of the subsequent servers would receive the information.”**

In the second paragraph of the Examiner’s comments, the Office states that

**“Each time a subsequent server receives the information, the information is a inbound payload to that server. Since all the payloads received by the servers are the same, each of the payloads are identical”**

Lastly, the Examiner states that the recitation “when OR IF” allowed for an interpretation wherein no later arriving outbound payload is received.

It is respectfully submitted that the amendments to the independent claims herewith distinguish the claimed embodiments from Mockapetris.

### **Independent Claim 1**

Claim 1 recites:

... each of the at least one gaming machine being configured to play at least one game and to carry out a game transaction for each game played and to commit each game transaction to each of the at least two central servers by sending a separate instance of a single transaction packet from the at least one gaming machine to each of the at least two central servers, each separate instance of the single transaction packet sent to each of the at least two central servers including an identical inbound game payload wherein each of the at least two central servers, upon receipt of the inbound game payload from the gaming machine having sent the instance of the transaction packet, are is configured to return send a single outbound game payload to the gaming machine having sent the instance of the transaction packet, the outbound game payload enabling the gaming machine having sent the instance of the transaction packet to complete the game transaction and wherein the at least one gaming machine is configured such that a first arriving outbound payload received by the at least one gaming machine is effective to complete the game transaction, irrespective of when ~~and~~ if a second later arriving outbound payload is received by the at least one gaming machine.

In particular, kindly note that each gaming machine is recited to be configured to send a separate instance of a single transaction packet from the gaming machine to each of at least two central servers. Thereafter, upon receipt of the inbound game payload from the gaming machine having sent the instance of the transaction packet, each of the at least two central servers is configured to send a single outbound game payload to the gaming machine having sent the instance of the transaction packet.

Mockapetris' "software ring" does not do this. In Mockapetris, "the source transmits the message to the first destination, which then forwards the message to the next destination, etc." Nowhere does Mockapetris teach or suggest that the source sends a separate instance of a single transaction packet to each of two destinations. In Mockapetris, the source sends a single instance of the message to a single destination. That destination (not the source) then forwards that message to the next destination.

Notwithstanding the Office's comment that "but for Mockapetris starting the process, none of the subsequent servers would receive the information", the claimed embodiment specifically requires the that each gaming machine send a separate instance of a single transaction packet from the gaming machine to each of at least two central servers. Mockapetris does not do that.

On the return side, the claim requires that "each of the at least two central servers, upon receipt of the inbound game payload from the gaming machine having sent the instance of the transaction packet, ~~are~~ is configured to ~~return~~ send a single outbound game payload to the gaming machine having sent the instance of the transaction packet". Therefore, the inbound game payload must be received "from the gaming machine having sent the instance of the transaction packet" for the server to send a single outbound payload to the gaming machine having sent the instance of the transaction packet. Therefore, the claim requires that the central server a) receive the instance of the transaction packet from the gaming machine having sent it, and b) send a single outbound game payload to the gaming machine having sent the instance of the transaction packet. This does not occur in Mockapetris, in which servers pass messages among themselves and the only the last destination returns the ACK to the sender. However, the last destination did not receive the message from the source (as is required in claim 1), but from the penultimate destination in the software ring. Therefore, it is respectfully submitted that claim 1 distinguishes over Mockapetris.

### **Independent Claim 79**

Claim 79 recites that each gaming machine is configured to

... carry out a game transaction for each game played and to commit each game transaction to each of the at least two central servers by sending a separate instance of a single transaction packet from the at least one gaming machine to each of the at least two central servers, each separate instance of the single transaction packet sent to each of the at least two central servers including an identical inbound game payload, wherein each

of the two central servers are configured such that any transaction packet that is not acknowledged by a non-responding one of the at least two central servers is sent directly from the synchronization engine of a responding one of the at least two central servers to the synchronization engine of the non-responding central server.

Here again, we have a gaming machine sending a separate instance of a single transaction packet from the gaming machine to each of at least two central servers. Therefore, Mockapetris' scheme of sending a message to a first server that then forwards the message to a second server does not meet the requirements of this claim, since the "source" only sends a single instance of the message to a single "destination". It is then that destination (and not the gaming machine, as required by the claim) that then sends (forwards) the message to a second "destination" in the ring.

#### **Independent Claim 108**

Claim 108, as amended, recites:

... committing each game transaction to each of the at least two central servers by sending a separate instance of a single transaction packet from the at least one gaming machine to each of the at least two central servers, each separate instance of the single transaction packet sent to each of the at least two central servers including an identical inbound game payload;

returning, by each of the at least two central servers, a single outbound game payload to the gaming machine upon receipt of the inbound game payload from the gaming machine having sent the instance of the transaction packet, and

Here again, the gaming machine is recited to send a separate instance of a single transaction packet from the gaming machine to each of the at least two central servers and returning, by each of these at least two central servers, a single outbound game payload upon receipt of the inbound game payload from the gaming machine having sent the instance of the transaction packet. Therefore, the claim requires that the central servers return a single outbound game payload to the gaming machine (and not to another server for forwarding to the gaming machine, as in

Mockapetris), upon receiving the inbound game payload from the gaming machine having sent the instance of the transaction packet (and not upon receipt of an inbound game payload from another server in the ring of servers, as in Mockapetris). It is believed that the claim amendments to this and the other independent claims foreclose any interpretation of the claim which would be consistent with Mockapetris.

### **Independent Claim 109**

Claim 109 recites, as amended:

committing each game transaction to each of the at least two central servers by sending, by the gaming machine, a separate instance of a single transaction packet from the gaming machine to each of the at least two central servers, each separate instance of the single transaction packet sent from the gaming machine to each of the at least two central servers including an identical inbound game payload;

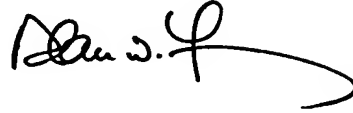
Here again, the game transaction is committed to the at least two central servers by the gaming machine sending a separate instance of a single transaction packet from the gaming machine to each of the at least two central servers. Mockapetris does not teach or suggest this and this embodiment is not taught or suggest in Mockapetris' "software ring".

None of these algorithms teach or suggest, whether considered alone or in combination with Nguyen (which teaches gaming machines), the claimed embodiments. Independent claim 109 is similarly not believed to be obvious over the Mockapetris-Nguyen combination, whether considered alone or in further combination with San Andres. Reconsideration and withdrawal of the outstanding rejections are, therefore, respectfully requested.

Applicants' attorney believes that the present application is now in condition for allowance and passage to issue. If any unresolved issues remain, the Examiner is respectfully invited to

contact the undersigned attorney of record at the telephone number indicated below, and whatever is required will be done at once.

Respectfully submitted,



Date: March 14, 2011

By: \_\_\_\_\_

Alan W. Young  
Attorney for Applicants  
Registration No. 37,970

YOUNG LAW FIRM, P.C.  
4370 Alpine Rd., Ste. 106  
Portola Valley, CA 94028  
Tel.: (650) 851-7210  
Fax: (650) 851-7232

C:\YLF\CLIENTS\CYBS\5872 (Univ Gaming Server)\5872 Amendment Filed Concurrently with RCE of 03-14-2011.doc